IFT 3913 – Qualité du logiciel et métriques UdeM/DIRO, Hiver 2015
Travail pratique #2

1. Introduction
Extraction de métriques à partir de diagrammes de classes UML (voir syntaxe du TP1).

2. Conditions de réalisation
Mêmes groupes que le TP1. Travail à remettre au plus tard lundi, le 30 mars lors de la séance de démo pour les rapports et au plus tard à 23h59 pour le code via le programme 'remise'.
Aucun retard ne sera accepté.

3. Travail à réaliser
En utilisant le programme Java qui permet de parser les diagrammes du TP1, ajouter une extension qui permet de calculer les métriques décrites dans la section 5. Le programme doit notamment permettre de
• Visualiser le résultat du calcul (dans l'interface usager).
• Accéder, à la demande, à la définition d'une métrique choisie par l'utilisateur.
• Produire à la demande un fichier CSV (séparation par des virgules) contenant pour un schéma, une matrice ayant pour lignes les classes et pour colonnes les métriques. Chaque cellule donne la valeur d'une métrique pour une classe

4. Barème
1. Rapport de 3 à 5 pages incluant la conception du programme (le diagramme de classes détaillé TP1 et TP2 plus les commentaires) 20% et le manuel d'utilisation 10%. La qualité du français sera prise en compte dans la correction.
2. Interface usager 20%.
3. Logique (programme) 50%.

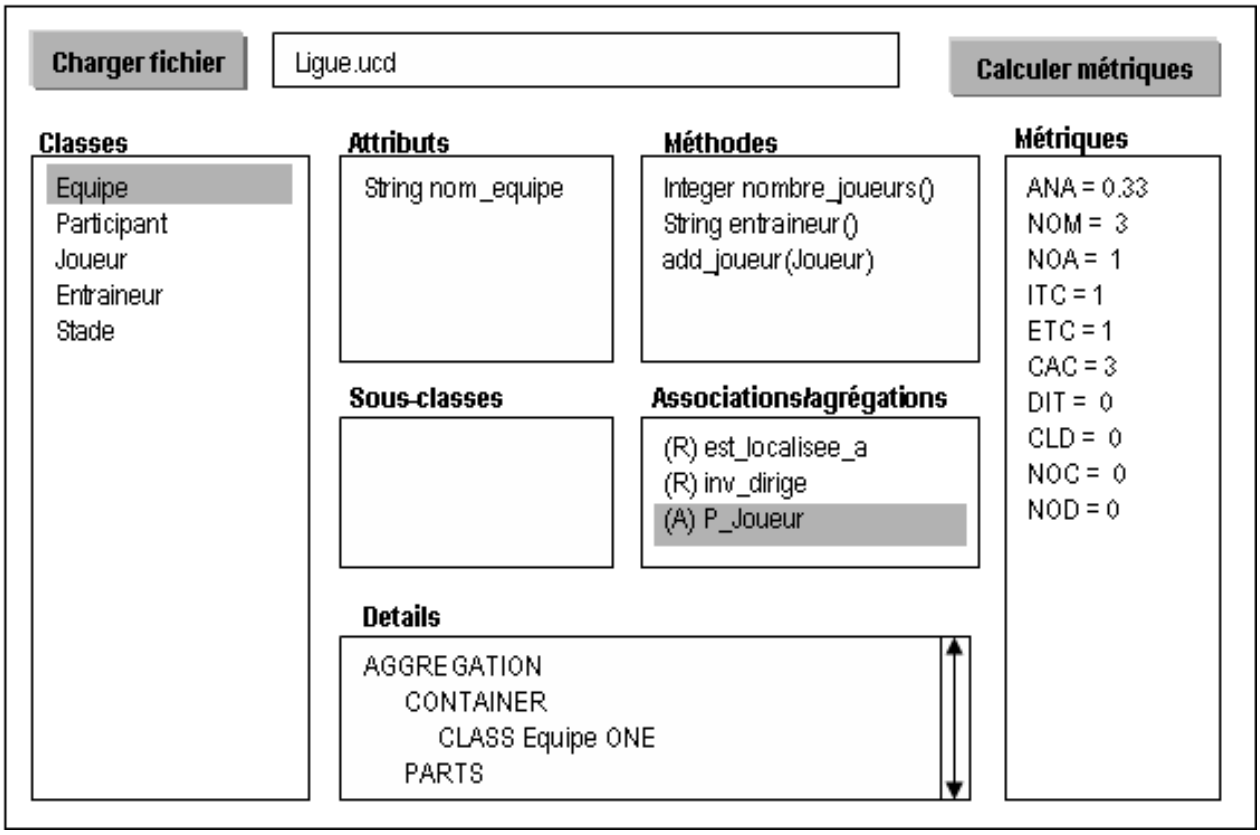
5. Description

Les métriques à calculer pour une classe ci sont

1. ANA(ci) : Nombre moyen d'arguments des méthodes locales pour la classe ci.
2. NOM(ci) : Nombre de méthodes locales/héritées de la classe ci. Dans le cas où une méthode est héritée et redéfinie localement (même nom, même ordre et types des arguments et même type de retour), elle ne compte qu'une fois.
3. NOA(ci) : Nombre d'attributs locaux/hérités de la classe ci.
4. ITC(ci) : Nombre de fois où d'autres classes du diagramme apparaissent comme types des arguments des méthodes de ci.
5. ETC(ci) : Nombre de fois où ci apparaît comme type des arguments dans les méthodes des autres classes du diagramme.
6. CAC(ci) : Nombre d'associations (incluant les agrégations) locales/héritées auxquelles participe une classe ci.
7. DIT(ci) : Taille du chemin le plus long reliant une classe ci à une classe racine dans le graphe d'héritage.
8. CLD(ci) : Taille du chemin le plus long reliant une classe ci à une classe feuille dans le graphe d'héritage.
9. NOC(ci) : Nombre de sous-classes directes de ci.
10. NOD(ci) : Nombre de sous-classes directes et indirectes de ci.

6. Exemple d'interface usager

La figure suivante donne, à titre d'indication seulement, un exemple d'interface graphique permettant de visualiser les résultats du parsing et du calcul de métriques.


Charger fichier
Ligue.ucd
Calculer métriques
Classes
Equipe
Participant
Joueur
Entraineur
Stade
Attributs
String nom_equipe
Méthodes
Integer nombre_joueurs()
String entraineur()
add_joueur(Joueur)
Métriques
ANA = 0.33
NOM = 3
NOA = 1
ITC = 1
ETC = 1
CAC = 3
DIT = 0
CLD = 0
NOC = 0
NOD = 0
Sous-classes
Associations/agrégations
(R) est_localisee_a
(R) inv_dirige
(A) P_Joueur
Details
AGGREGATION
CONTAINER
CLASS Equipe ONE
PARTS